Tales from DeCrypt

In newsgroups, lists, and other ways of publicly sharing information, one popular method of obscuring information without actually hiding it has been the ROT13 algorithm: alphabetic characters are simply rotated by 13 positions (modulo 26), so that the encryption and decryption algorithms are identical. Messages that are potentially offensive to some readers of the newsgroup or list are purposely posted in ROT13 form, on the theory that the **reader** is responsible for changing the offensive material into clear-text form, and so that the reader cannot complain about it.

We can use a rotational cipher to hide information as well as obscure it. You are responsible for generating the decryption algorithm for the encryption algorithm described here. It is restricted to 7-bit ASCII/ANSI characters, and we will deal only with the printing characters — 0x20 (space) up to and including 0x7e ('~') for 95 characters. This way the encrypted text can still be dealt with as pure text for file manipulation and transmission purposes.

Encryption Algorithm

Select three numbers to encode all the necessary information for this linear congruential random number generator: the multiplier (`a`), the modulus (`m`), and the seed (`s`):

```
double r(in int a, in int m, inout int s)
   double val = s modulo m / double(m)
   s = ( a * s + 1 ) modulo m
   return val
```

The two integers for the random number generator and the initial seed (`s` in the pseudo code above) are contained within the file as the first line, which contains three white-space delimited 32-bit integers, given in the order "`a m s`". Range: $2 \leq$ number ≤ 65536. So a first line of "`12343 65536 11`" generates

- `a = 12343`
- `m = 65536`
- `s = 11`

The output file of the encryption program contains these three numbers, white-space delimited, on one line. The encrypted text begins on the next line. This constitutes the input to your decryption program.

Encrypted text is generated by the following algorithm:

```
for each character c in the input stream
    if the character is not in the range 0x20 through 0x7e
        pass it through to the output
    otherwise
        c = ( (c - 32) + ceiling( 95 - r(a,m,seed)*95) ) ) modulo 95 + 32
        send c to the output
```

Input

The input for your program is the output of the encryption program: three white-space delimited numbers on one line. The encrypted text begins on the next line, and continues to the end of file.

Output

Your output is the decryption of the text encrypted in the input file.

Sample Input	**Sample Output**
12343 65536 11 a[+d'x/vKmV<WP(+2:N]%CN+u#rjNQB vW'ecvzcK5E%F;^Qlo~pt\]kwGr*.yv| So|#p36LhPNM#"N<I|}2c[cGX5I3o!u m48rOK1&N=&8%Q-2Jq^[v&r;at"z#'C	aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb ccccccccccccccccccccccccccccc ddddddddddddddddddddddddddddd